

Learning convolutional neural network to maximize Pos@Top performance measure

Ru-Ze Liang
King Abdullah University of
Science and Technology
Thuwal 23955, Saudi Arabia
ruzeliang@outlook.com

Gaoyuan Liang
Jiangsu University of
Technology
Jiangsu 213001, China
gaoyuanliang@outlook.com

Weizhi Li
Suning Commerce R&D
Center USA, Inc.
Palo Alto, CA 94304, USA

Yi Gu
Analytics & Research,
Travelers Companies Inc.
Hartford, CT 06183, USA

Qinfeng Li
Hohai University
Nanjing 210098, China

Jim Jing-Yan Wang
New York University Abu
Dhabi
Abu Dhabi, United Arab
Emirates

ABSTRACT

In the machine learning problems, the performance measure is used to evaluate the machine learning models. Recently, the number positive data points ranked at the top positions (Pos@Top) has been a popular performance measure in the machine learning community. In this paper, we propose to learn a convolutional neural network (CNN) model to maximize the Pos@Top performance measure. The CNN model is used to represent the multi-instance data point, and a classifier function is used to predict the label from the its CNN representation. We propose to minimize the loss function of Pos@Top over a training set to learn the filters of CNN and the classifier parameter. The classifier parameter vector is solved by the Lagrange multiplier method, and the filters are updated by the gradient descent method alternately in an iterative algorithm. Experiments over benchmark data sets show that the proposed method outperforms the state-of-the-art Pos@Top maximization methods.

Keywords

Convolutional neural network; Multi-instance learning; Multivariate performance measure; Positive at top

1. INTRODUCTION

In the machine learning and data mining applications, the performance measures are used to evaluate the performance

of the predictive models [20, 19, 5]. The outputs of the predictive model over a set of test data points are compared to their ground truth labels, and the performance measures are used to produce the performance scores. The performance measures include the area under the receiver operating characteristic curve (AUC), the recall-precision break-even point (RPB), the top k -rank precision (Top k Pre), and the positives at top (Pos@Top) [14, 21, 22, 27, 7]. Recently the performance of Pos@Top is being more and more popular in the machine learning community. This performance measures only counts the positive instances ranked before the first-ranked negative instance. The rate of these positive instances in all the positive instances is defined as the **Pos@Top**. In many machine learning applications, we observed that the top-ranked instances/classes plays critical roles in the performance evaluation, and the Pos@Top performance measure can give a good description about how the top-ranked instances/classes distribute [2, 4, 16]. Moreover, it is parameter free, and it use the top-ranked negative as the boundary of the recognized positive instance pool.

Although this performance measure has been used in various applications, its usage is limited to the test process, but is ignored in the training process. This problem can result in a classifier not optimal for the maximization of the Pos@Top performance measure. To solve this problem, a few works were proposed to optimize the Pos@Top measure in the training process directly. Li et al. [16] proposed a highly efficient algorithm to maximize the Pos@Top over the training set by learning a linear classifier model, and named it as TopPush. This algorithm has a linear time-complexity with regard to the number of the training instances. Agarwal [3] proposed a ranking algorithm which can maximize the Pos@Top performance measure over the training set. It is a support vector style model, but it has a different objective, and it is solved not as a quadratic programming problem, but as a $\ell_{1,\infty}$ constrained problem. Boyd et al. [4] proposed to maximize the convex surrogate of the loss function of Pos@Top, which is the number of positive data instances ranked behind the too-ranked negative instances. The objective is optimized as a set of convex optimization problems. Among these three existing works of optimization

of the Pos@Top, all the predictive models are linear models, and they are designed to tickle the single-instance data.

To solve these problems, we develop a convolutional neural network (CNN) model to optimize the Pos@Top performance measure. This model is designed to tickle the multiple instance sequence as input. It is composed convolutional layer, the activation layer, the max-pooling. Moreover, the output of the CNN is used as a representation of the multi-instance data point, and we apply a linear classifier to predict the class label. We propose to learn the parameters of the CNN and classifier model, including the filters and the classifier parameter to maximize the Pos@Top. To this end, we argue that the loss function should be defined as the number of the positive instances sorted behind the top-ranked negative instance. We define a hinge loss function for this problem. We solve this problem by alternate optimization problem and develop an iterative algorithm.

The following parts of the paper are organized as follows. In section 2, we introduce the proposed classification model and the learning method of the parameters of the model. In section 3, we evaluate the proposed method over some benchmark data sets. In section 4, the conclusion of this paper is summarized.

2. PROPOSED METHOD

2.1 Problem modeling

The training set is composed of n data points, and denoted as $\{(X_i, y_i)\}_{i=1}^n$, where X_i is the input data of the i -th data point, and $y_i \in \{+1, -1\}$ is its binary class label. X_i is a sequence of m_i instances, denoted as a matrix $X_i = [\mathbf{x}_{i1}, \dots, \mathbf{x}_{im_i}] \in \mathbb{R}^{d \times m_i}$, where $\mathbf{x}_{i\kappa} \in \mathbb{R}^d$ is the d -dimensional feature vector of the κ -th instance. To represent the i -th data point, we use a CNN model,

$$\mathbf{g}(X_i) = \max(\phi(W^\top X_i)) \in \mathbb{R}^m. \quad (1)$$

In this model, $W = [\mathbf{w}_1, \dots, \mathbf{w}_m] \in \mathbb{R}^{d \times m}$ is the matrix of m filters, and $\mathbf{w}_k \in \mathbb{R}^d$ is the k -th filter vector. $\phi(\cdot)$ is a element-wise non-linear activation function, defined as $\phi(x) = \frac{1}{1 + \exp(-x)}$. $\max(\cdot)$ is the row-wise maximization operator, and it selects the maximum element from each row of a matrix. To approximate the class label y_i of a data point X_i from its CNN representation $\mathbf{g}(X_i)$, we propose to use a linear classifier,

$$y_i \leftarrow f(X_i) = \mathbf{u}^\top \mathbf{g}(X_i). \quad (2)$$

$\mathbf{u} \in \mathbb{R}^m$ is a parameter vector of the linear classifier. The overall framework of the proposed model is shown in Figure 1.

The argued performance measure, Pos@Top, is defined as number of positive data points which are ranked before the top-ranked negative data point, $\max_{j: y_j = -1} f(X_j)$. To maximize the Pos@Top, we argue a 0-1 loss function for each positive data point,

$$\ell_{0-1}(X_i, y_i) = \mathcal{I} \left(f(X_i) \leq \max_{j: y_j = -1} f(X_j) \right), \forall i: y_i = +1. \quad (3)$$

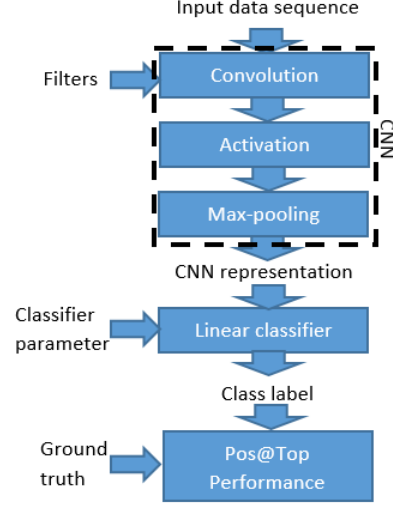


Figure 1: Overall CNN Pos@Top maximization framework.

To minimize this loss function, we argue that for any positive data point, its classification score should be larger than that of the top-ranked negative plus a margin amount,

$$f(X_i) > \max_{j: y_j = -1} f(X_j) + 1, \forall i: y_i = +1, \quad (4)$$

we further propose a hinge loss function as follows to give a loss when this condition does not hold,

$$\ell_{hinge}(X_i, y_i) = \max \left(0, \max_{j: y_j = -1} f(X_j) - f(X_i) + 1 \right), \quad (5)$$

$$\forall i: y_i = +1.$$

To learn the CNN classifier parameters W and \mathbf{u} to maximize the Pos@Top, we should minimize the loss function of (5) over all the positive data points. Mean while we also propose to regularize the filter parameters and the full connection weights to prevent the over-fitting problem, and the squared ℓ_2 norms of \mathbf{u} and W are minimized. The overall objective function of the learning problem is the combination of the losses measured by (5) over the positive data points, the regularization terms of \mathbf{u} and W , and the minimization problem is given as follows,

$$\min_{W, \mathbf{u}} \left\{ \frac{1}{2} \|\mathbf{u}\|_2^2 + C_1 \sum_{i: y_i = +1} \max \left(0, \max_{j: y_j = -1} f(X_j) - f(X_i) + 1 \right) + C_2 \|W\|_2^2 \right\}. \quad (6)$$

where C_1 and C_2 are the tradeoff parameters. We further define θ as the classification score of the top negative, and ξ_i as the response of (5),

$$\begin{aligned}
\theta &= \max_{j: y_j = -1} f(X_j), \text{ and} \\
\xi_i &= \max \left(0, \max_{j: y_j = -1} f(X_j) - f(X_i) + 1 \right) \\
&= \max(0, \theta - f(X_i) + 1).
\end{aligned} \tag{7}$$

With the slack variables, we rewrite the problem in (6) as follows,

$$\begin{aligned}
\min_{W, \mathbf{u}, \xi_i | i, y_i = +1, \theta} & \left\{ \frac{1}{2} \|\mathbf{u}\|_2^2 + C_1 \sum_{i: y_i = +1} \xi_i + C_2 \|W\|_2^2 \right\}, \\
s.t. & \forall j, y_j = -1 : f(X_j) \leq \theta, \\
& \forall i, y_i = +1 : 0 \leq \xi_i, \text{ and } \theta - f(X_i) + 1 \leq \xi_i.
\end{aligned} \tag{8}$$

2.2 Problem optimization

The dual form of the optimization problem is as follows

$$\begin{aligned}
\max_{\boldsymbol{\delta}} \min_W & \left\{ -\frac{1}{2} \sum_{i, i'=1}^n \delta_i \delta_{i'} y_i y_{i'} \mathbf{g}(X_i)^\top \mathbf{g}(X_{i'}) \right. \\
& + C_2 \|W\|_2^2 + \boldsymbol{\epsilon}^\top \boldsymbol{\delta} \\
& = -\frac{1}{2} \sum_{i, i'=1}^n \delta_i \delta_{i'} y_i y_{i'} \sum_{k=1}^m \phi(\mathbf{w}_k^\top \mathbf{x}_{i\psi_{ik}}) \phi(\mathbf{w}_k^\top \mathbf{x}_{i'\psi_{i'k}}) \\
& \left. + C_2 \|W\|_2^2 + \boldsymbol{\epsilon}^\top \boldsymbol{\delta} \right\}, \\
s.t. & \boldsymbol{\delta} \geq 0, \text{diag}(\boldsymbol{\epsilon})\boldsymbol{\delta} \leq C_1 \mathbf{1}, \mathbf{y}^\top \boldsymbol{\delta} = 0,
\end{aligned} \tag{9}$$

where $\boldsymbol{\epsilon} = [\epsilon_1, \dots, \epsilon_n] \in \{1, 0\}^n$ and $\epsilon_i = 1$ if $y_i = 1$, and 0 otherwise. δ_i is the Lagrange multiplier variable for the constraint $\theta - f(X_i) + 1 \leq \xi_i$ if $y_i = +1$, and that for the constraint $f(X_i) \leq \theta$ otherwise. $\boldsymbol{\delta} = [\delta_1, \dots, \delta_n]^\top$, $\mathbf{y} = [y_1, \dots, y_n]^\top$. ψ_{ik} is defined as

$$\psi_{ik} = \arg \max_{\kappa=1}^{m_i} \phi(\mathbf{w}_k^\top \mathbf{x}_{i\kappa}), \tag{10}$$

and it indicates the instance in a bag which gives the maximum response with regard to a filter. We propose to use an alternate optimization strategy to optimize this problem. In an iterative algorithm, $\boldsymbol{\delta}$ and W are updated alternately.

2.2.1 Updating $\boldsymbol{\delta}$

To optimize $\boldsymbol{\delta}$, we fix W and remove the irrelevant term to obtain the following optimization problem,

$$\begin{aligned}
\max_{\boldsymbol{\delta}} & \left\{ -\frac{1}{2} \sum_{i, i'=1}^n \delta_i \delta_{i'} y_i y_{i'} \sum_{k=1}^m \phi(\mathbf{w}_k^\top \mathbf{x}_{i\psi_{ik}}) \phi(\mathbf{w}_k^\top \mathbf{x}_{i'\psi_{i'k}}) \right. \\
& \left. + \boldsymbol{\epsilon}^\top \boldsymbol{\delta} \right\}, \\
s.t. & \boldsymbol{\delta} \geq 0, \text{diag}(\boldsymbol{\epsilon})\boldsymbol{\delta} \leq C_1 \mathbf{1}, \mathbf{y}^\top \boldsymbol{\delta} = 0,
\end{aligned} \tag{11}$$

This is a linear constrained quadratic programming problem, and we can use an active set algorithm to solve it. After it is solved, we can recover \mathbf{u} as follows,

$$\mathbf{u} = \sum_{i=1}^n \delta_i y_i \mathbf{g}(X_i). \tag{12}$$

2.2.2 Updating W

To optimize W , we fix $\boldsymbol{\delta}$ and remove the irrelevant terms to obtain the following problem,

$$\begin{aligned}
\min_W & \left\{ -\frac{1}{2} \sum_{i, i'=1}^n \delta_i \delta_{i'} y_i y_{i'} \sum_{k=1}^m \phi(\mathbf{w}_k^\top \mathbf{x}_{i\psi_{ik}}) \phi(\mathbf{w}_k^\top \mathbf{x}_{i'\psi_{i'k}}) \right. \\
& \left. + C_2 \|W\|_2^2 = \sum_{k=1}^m s(\mathbf{w}_k) \right\}
\end{aligned} \tag{13}$$

where

$$\begin{aligned}
s(\mathbf{w}_k) &= -\frac{1}{2} \sum_{i, i'=1}^n \delta_i \delta_{i'} y_i y_{i'} \phi(\mathbf{w}_k^\top \mathbf{x}_{i\psi_{ik}}) \phi(\mathbf{w}_k^\top \mathbf{x}_{i'\psi_{i'k}}) \\
& + C_2 \|\mathbf{w}_k\|_2^2.
\end{aligned} \tag{14}$$

It is clear that $s(\mathbf{w}_k)$ is a independent function of \mathbf{w}_k , thus we can update the filters one by one. When one filter is being updated, other filters are fixed. The updating of \mathbf{w}_k is conducted by gradient descent,

$$\mathbf{w}_k \leftarrow \mathbf{w}_k - \eta \nabla s_{\mathbf{w}_k}(\mathbf{w}_k), \tag{15}$$

and the gradient function of $s_{\mathbf{w}_k}(\mathbf{w}_k)$ is calculated as follows,

$$\begin{aligned}
\nabla s(\mathbf{w}_k) &= -\frac{1}{2} \sum_{i, i'=1}^n \delta_i \delta_{i'} y_i y_{i'} \left(\frac{\exp(-\mathbf{w}_k^\top \mathbf{x}_{i\psi_{ik}}) \mathbf{x}_{i\psi_{ik}}}{(1 + \exp(-\mathbf{w}_k^\top \mathbf{x}_{i\psi_{ik}}))^2} \right. \\
& \left. \phi(\mathbf{w}_k^\top \mathbf{x}_{i'\psi_{i'k}}) + \phi(\mathbf{w}_k^\top \mathbf{x}_{i\psi_{ik}}) \frac{\exp(-\mathbf{w}_k^\top \mathbf{x}_{i'\psi_{i'k}}) \mathbf{x}_{i'\psi_{i'k}}}{(1 + \exp(-\mathbf{w}_k^\top \mathbf{x}_{i'\psi_{i'k}}))^2} \right) \\
& + C_2 \mathbf{w}_k.
\end{aligned} \tag{16}$$

2.3 Iterative algorithm

Based on the optimization results of section 2.2, we develop an iterative algorithm to learn both W and \mathbf{u} . The algorithm is given as in Algorithm 1. In this algorithm, we repeat the iterations until convergence. In each iteration, we first update the indicator of the maximally responding the filters, then update the Lagrange multipliers, and finally update the filters. The proposed algorithm is named as Convolutional Max Pos@Top Classifier (ConvMPT).

3. EXPERIMENTS

In this section of experiments, we evaluate the proposed algorithm over several multiple instance data set for the problem of maximization of Pos@Top.

Algorithm 1 Iterative learning algorithm for CNN based Pos@Top maximization.

Input: $\{(X_i, y_i)\}_{i=1}^n$.

Initialize W randomly.

repeat

if $i = 1, \dots, n, k = 1, \dots, m$ **then**

 Update $\psi_{ik} = \arg \max_{\kappa=1}^m \phi(\mathbf{w}_k^\top \mathbf{x}_{i\kappa})$.

end if

 Update δ by solving the following quadratic programming problem,

$$\begin{aligned} \delta^* = & \arg \max_{\delta} \left\{ -\frac{1}{2} \sum_{i,i'=1}^n \delta_i \delta_{i'} y_i y_{i'} \sum_{k=1}^m \phi(\mathbf{w}_k \mathbf{x}_{i\psi_{ik}}) \phi(\mathbf{w}_k \mathbf{x}_{i'\psi_{i'k}}) \right. \\ & \left. + \epsilon^\top \delta \right\}, \\ & s.t. \delta \geq 0, \text{diag}(\epsilon)\delta \leq C_1 \mathbf{1}, \mathbf{y}^\top \delta = 0, \end{aligned} \quad (17)$$

for $K = 1, \dots, m$ **do**

 Calculate the gradient function $\nabla s(\mathbf{w}_k)$ as in (16).

 Update the k -th filter, $\mathbf{w}_k \leftarrow \mathbf{w}_k - \eta \nabla s_{\mathbf{w}_k}(\mathbf{w}_k)$.

end for

until Convergence

Calculate the classifier parameter $\mathbf{u} = \sum_{i=1}^n \delta_i y_i \mathbf{g}(X_i)$.

Output: W and \mathbf{u} .

3.1 Experimental data sets and setup

In the experiment, we use tree types of data set — image set, text set, and audio set.

- The image set used by us is the Caltech-256 dataset [11]. In this set, we have 30,607 images in total. These images belongs to 257 classes. Each image is presented as a bag of multiple instances. To this end, each image is split into a group of small image patches, and each patch is an instance.
- The text data set used in this experiment is the Semeval-2010 task 8 data set [13]. It contains 10,717 sentences, and these sentences belongs to 10 different classes of relations. Each sentence is composed of several words, and thus it is natural a multiple instance data set. Each word is represented as a feature vector of 100 dimensions using the word embedding algorithm proposed by Sun et al. [24].
- The audio data set used in this experiment is the Spoken Arabic digits (SAD) [12]. In this data set, there are 8,800 sequences of voice signal. These voice signal sequences belongs to 10 classes of digits. To represent each sequence, we split it to a group of voice signal frames. Each frame is an instance, thus each sequence is a bag of multiple instances.

The summary of the information of the data sets are listed in the Table 1.

Table 1: Information of the data sets used in the experiments.

Data set	Instance type	# bags	# classes
Caltech-256	Image patch	30,607	257
Semeval-2010 task 8	Word	10,717	10
SAD	Voice frame	8,800	10

To perform the experiments over the data sets, we use the 10-fold cross-validation protocol to split training and test sets. The values of the tradeoff parameters of the ConvMPT are chosen by the cross-validation over the training set in the experiments, and the average Pos@Top values over the test sets are reported as the results. We use a one-vs-all strategy for the multi-class problem.

3.2 Experimental results

- We firstly compare the proposed algorithm against ordinary convolutional network learning method. We use the TensorFlow as the implementation of the ordinary convolutional network model [1], and it logistic loss function as measure of performance measure. The results are reported in Table 2 and Figure 2. The results clearly show that the proposed method outperforms the ordinary CNN significantly over all the three data sets of different types.

Table 2: Results of comparison to ordinary convolutional network learning method.

Methods	Caltech-256	Semeval-2010 task 8	SAD
ConvMPT	0.268	0.462	0.510
Ordinary CNN	0.201	0.368	0.437

Results of comparison to ordinary convolutional network learning method.

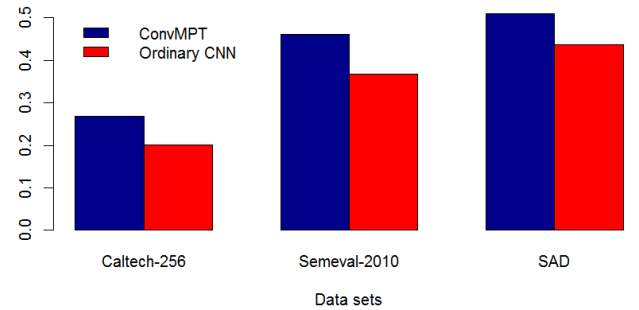


Figure 2: Results of comparison to ordinary convolutional network learning method.

- We also compare the proposed method against some other algorithms for optimization of the Pos@Top performance. The compared methods are TopPush proposed by Li et al. [16], AATP proposed by Boyd et al. [4], and InfinitePush proposed by Agarwal [2]. The comparison results are shown in Table 3 and Figure 3.

According to the results reported in the table, the proposed convolutional Pos@Top maximizer outperforms the other methods. The compared methods, although used different optimization methods, but all aims to optimize a linear classifier, but our model has a convolutional structure. The results also show that convolutional network is a good choice for the problem of optimization of Pos@Top.

Table 3: Results of comparison to existing Pos@Top maximization method.

Methods	Caltech-256	Semeval-2010 task 8	SAD
ConvMPT	0.268	0.462	0.510
TopPush	0.239	0.401	0.497
AATP	0.213	0.387	0.488
InfinitePush	0.211	0.370	0.476

Results of comparison to existing Pos@Top maximization method.

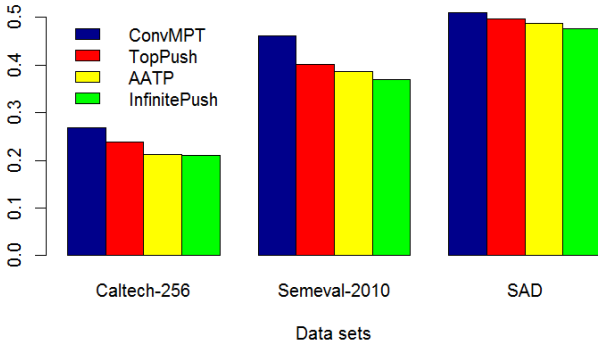


Figure 3: Results of comparison to existing Pos@Top maximization method.

- The algorithm is an iterative algorithm, so we also study how the algorithm performs with different numbers of iterations. We report the average of rate of Pos@Top with regard to different iterations in Figure 4. From this table, we can observe a trend of improving performance with growing number of iterations. But generally speaking, the algorithm is stable to the change of iteration numbers.

4. CONCLUSION

In this paper, we propose a novel model to maximize the performance of Pos@Top. The proposed model has a structure of CNN. The parameter learning of CNN is to optimize the loss function of Pos@Top. We propose a novel iterative learning algorithm to solve this problem. Meanwhile we also propose to minimize the squared ℓ_2 norm of the filter matrix of the convolutional layer. The proposed algorithm is compared to the ordinary CNN and the existing Pos@Top minimization method, and the results show its advantage. In the future, we will apply the proposed method to other

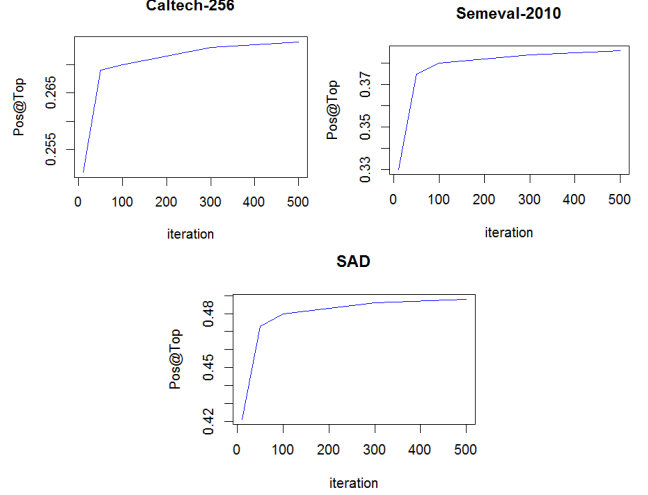


Figure 4: Results of comparison to existing Pos@Top maximization method.

applications, such as medical imaging [18, 23, 15, 25, 17], computer vision [26], network security [29, 28, 31, 30], et al. We will also consider learning Bayesian network to maximize the Pos@Top performance measure [10, 6, 10, 9, 8].

5. REFERENCES

- [1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous systems, 2015. *Software available from tensorflow.org*, 1, 2015.
- [2] S. Agarwal. The infinite push: A new support vector ranking algorithm that directly optimizes accuracy at the absolute top of the list. In *Proceedings of the 11th SIAM International Conference on Data Mining, SDM 2011*, pages 839–850, 2011.
- [3] N. S. Al Madi and J. I. Khan. Measuring learning performance and cognitive activity during multimodal comprehension. In *2016 7th International Conference on Information and Communication Systems (ICICS)*, pages 50–55. IEEE, 2016.
- [4] S. Boyd, C. Cortes, M. Mohri, and A. Radovanovic. Accuracy at the top. In *Advances in Neural Information Processing Systems*, volume 2, pages 953–961, 2012.
- [5] J. Fan and R.-Z. Liang. Stochastic learning of multi-instance dictionary for earth mover’s distance-based histogram comparison. *Neural Computing and Applications*, pages 1–11, 2016.
- [6] X. Fan, B. Malone, and C. Yuan. Finding optimal bayesian network structures with constraints learned from data. In *Proceed. of the 30th Conf. on Uncertainty in Artificial Intelligence (UAI-2014)*, 2014.
- [7] X. Fan and K. Tang. Enhanced maximum auc linear classifier. In *Fuzzy Systems and Knowledge Discovery (FSKD), 2010 Seventh International Conference on*,

- volume 4, pages 1540–1544. IEEE, 2010.
- [8] X. Fan, K. Tang, and T. Weise. Margin-based over-sampling method for learning from imbalanced datasets. In *Proceedings of the 15th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-2011)*, pages 309–320. Springer Berlin Heidelberg, 2011.
- [9] X. Fan and C. Yuan. An improved lower bound for bayesian network structure learning. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI-2015)*, 2015.
- [10] X. Fan, C. Yuan, and B. Malone. Tightening bounds for bayesian network structure learning. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI-2014)*, 2014.
- [11] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. 2007.
- [12] N. Hammami, M. Bedda, and N. Farah. Spoken arabic digits recognition using mfcc based on gmm. In *Sustainable Utilization and Development in Engineering and Technology (STUDENT), 2012 IEEE Conference on*, pages 160–163. IEEE, 2012.
- [13] I. Hendrickx, S. N. Kim, Z. Kozareva, P. Nakov, D. Ó Séaghdha, S. Padó, M. Pennacchiotti, L. Romano, and S. Szpakowicz. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, pages 94–99. Association for Computational Linguistics, 2009.
- [14] T. Joachims. A support vector method for multivariate performance measures. In *Proceedings of the 22nd international conference on Machine learning*, pages 377–384. ACM, 2005.
- [15] D. R. King, W. Li, J. J. Squiers, R. Mohan, E. Sellke, W. Mo, X. Zhang, W. Fan, J. M. DiMaio, and J. E. Thatcher. Surgical wound debridement sequentially characterized in a porcine burn model with multispectral imaging. *Burns*, 41(7):1478–1487, 2015.
- [16] N. Li, R. Jin, and Z.-H. Zhou. Top rank optimization in linear time. In *Advances in Neural Information Processing Systems*, volume 2, pages 1502–1510, 2014.
- [17] W. Li, W. Mo, X. Zhang, Y. Lu, J. J. Squiers, E. W. Sellke, W. Fan, J. M. DiMaio, and J. E. Thatcher. Burn injury diagnostic imaging device’s accuracy improved by outlier detection and removal. In *SPIE Defense+ Security*, pages 947206–947206. International Society for Optics and Photonics, 2015.
- [18] W. Li, W. Mo, X. Zhang, J. J. Squiers, Y. Lu, E. W. Sellke, W. Fan, J. M. DiMaio, and J. E. Thatcher. Outlier detection and removal improves accuracy of machine learning approach to multispectral burn diagnostic imaging. *Journal of biomedical optics*, 20(12):121305–121305, 2015.
- [19] R.-Z. Liang, L. Shi, H. Wang, J. Meng, J. J.-Y. Wang, Q. Sun, and Y. Gu. Optimizing top precision performance measure of content-based image retrieval by learning similarity function. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*. IEEE, 2016.
- [20] R.-Z. Liang, W. Xie, W. Li, H. Wang, J. J.-Y. Wang, and L. Taylor. A novel transfer learning method based on common space mapping and weighted domain matching. In *Tools with Artificial Intelligence (ICTAI), 2016 IEEE 28th International Conference on*. IEEE, 2016.
- [21] S. Lu, H. Lu, and W. J. Kolarik. Multivariate performance reliability prediction in real-time. *Reliability Engineering & System Safety*, 72(1):39–45, 2001.
- [22] Q. Mao and I. W.-H. Tsang. A feature selection method for multivariate performance measures. *IEEE transactions on pattern analysis and machine intelligence*, 35(9):2051–2063, 2013.
- [23] W. Mo, R. Mohan, W. Li, X. Zhang, E. W. Sellke, W. Fan, J. M. DiMaio, and J. E. Thatcher. The importance of illumination in a non-contact photoplethysmography imaging system for burn wound assessment. In *SPIE BiOS*, pages 93030M–93030M. International Society for Optics and Photonics, 2015.
- [24] F. Sun, J. Guo, Y. Lan, J. Xu, and X. Cheng. Learning word representations by jointly modeling syntagmatic and paradigmatic relations. volume 1, pages 136–145, 2015.
- [25] J. E. Thatcher, W. Li, Y. Rodriguez-Vaqueiro, J. J. Squiers, W. Mo, Y. Lu, K. D. Plant, E. Sellke, D. R. King, W. Fan, et al. Multispectral and photoplethysmography optical imaging techniques identify important tissue characteristics in an animal model of tangential burn excision. *Journal of Burn Care & Research*, 37(1):38–52, 2016.
- [26] C.-Y. Wang, D.-Y. Peng, L. Xu, and X.-S. Yi. Gradual gray-watermark embedding algorithm in the wavelet domain [j]. *Journal of Computer Applications*, 6:025, 2007.
- [27] J. J.-Y. Wang, I. W.-H. Tsang, and X. Gao. Optimizing multivariate performance measures from multi-view data. In *Thirtieth AAAI Conference on Artificial Intelligence*, pages 2152–2158, 2016.
- [28] L. Xu, Z. Zhan, S. Xu, and K. Ye. Cross-layer detection of malicious websites. In *Proceedings of the third ACM conference on Data and application security and privacy*, pages 141–152. ACM, 2013.
- [29] L. Xu, Z. Zhan, S. Xu, and K. Ye. An evasion and counter-evasion study in malicious websites detection. In *Communications and Network Security (CNS), 2014 IEEE Conference on*, pages 265–273. IEEE, 2014.
- [30] S. Xu, W. Lu, and L. Xu. Push-and pull-based epidemic spreading in networks: Thresholds and deeper insights. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 7(3):32, 2012.
- [31] S. Xu, W. Lu, L. Xu, and Z. Zhan. Adaptive epidemic dynamics in networks: Thresholds and control. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 8(4):19, 2014.